

In the Specification

Please replace paragraphs [0031] on pg. 11; [0034] on pg. 12; [0035] of page 13 and [0038] on page 14 with the following replacement paragraphs with changes. No new matter has been added.

[0031] As shown in Figure 2, client computing system 116a comprises OS 218a and application software 219a. Client computing system 116a further comprises client access software 220a. Generally, client access software 220a identifies input-related instructions such as mouse and keyboard inputs and processes the instructions in translator 222a. Translator 222a converts the native instruction format of OS 218a into an open standard format instruction represented by an XML element. The XML element is forwarded over network 114 to the one of server computing systems 112a-c, ~~with~~ which system 116a is remotely controlling. Client access software 220a also operates to receive XML elements representing output-related instructions from server computing system 112 via network 114. The XML formatted output-related instructions are processed by translator 222a to arrive ~~[[at]]~~ as instructions in a native format that can be processed by OS ~~218a~~ 219.

[0034] Client computing system 116c comprises OS 218c and application 219c, both of which operate generally as described above in connection with system 116a. In ~~server~~ client computing system 116c, remote access server 220c is integrated with application 219c, rather than with OS 218c. Thus application 219c in combination with remote access server 220c is operable to communicate over network 114 without interfering with OS 218c. Client computing system 116c does not comprise a translator. Rather, application 219c is operable to create commands directly in an open format such as, for example, XML data format. Therefore, there is no need to translate commands to and from application 219c's native format prior to and after transmission across network 114.

[0035] Figure 3 is a flow chart of an illustrative process for providing remote computer access. As shown, at step 310, a request to initiate a remote access session is received at one of the server computing systems 112, for example system 112a, from one of the client computing

systems 116, for example system 116a, via network 114. In an illustrative embodiment, the request is received and handled by the remote access server 214a. At step 312, remote access server 214a causes the desired application, which may be application 212a of Fig. 2, to be launched on system 112a. During execution of application 212a on OS 210, remote access server 214a recognizes, at step 314, an output-related instruction such as, for example, an instruction related to displaying data or generating a sound. At step 316, translator 216a translates the instruction from a native format for execution by OS 210a into an open system data item such as, for example, an XML element. XML is a markup language that is used to describe data and is known by those skilled in the art. While those skilled in the computing arts are knowledgeable regarding XML and can implement an XML system, background regarding implementing XML systems is disclosed at www.w3c.org, the contents of which are hereby incorporated by reference in their entirety. Generally, an XML element is defined by at least a beginning tag, a data item corresponding to the element content, and a closing tag. For example, a native OS 210 instruction for displaying a file (referred to as *filename*) may be translated into an XML element such as the following: `<display>filename</display>`. The tag “`<display>`” represents the beginning of the XML element, the tag “`</display>`” represents the end of the XML element, and *filename* corresponds to the data to be displayed. Similarly, a native OS 210 instruction for playing an audio file may be translated by translator 216 into an XML formatted data item, for example, such as the following: `<play>filename</play>`. Indeed, an XML equivalent may be developed for all native instructions. Translator 216 may maintain a database matching native instructions to corresponding XML elements. For example, an instruction to display a file may be matched to XML tags “`<display>`” and “`</display>`”. Upon receiving an output native instruction, translator 216 may access the database to identify the corresponding XML element(s). At step 316, OS 210a may also execute the native format instruction that is being translated so that the output is generated at server computing system 112 as well as at client computing system 116.

[0038] At step 324, the output instructions are executed. In the exemplary scenario where the remote access session is established from client computing system 116a, the instructions are executed by OS 218a 240a.